

## VARIABLE LENGTH DECODING SYSTEM AND METHOD

## CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority from provisional application No. 60/462,501, filed 04/11/2003 and is a continuation-in-part of pending application No. 09/788,807, filed 02/20/2001, *is now abandon*.  
1

## BACKGROUND OF THE INVENTION

The present invention relates to electronic systems, and more particularly, to digital systems and methods with bitstreams representing coded information with codewords of variable length.

The current rapid expansion of digital communication (speech, video, and data) relies on increasingly economical digital signal processing and efficient transmission and storage. For example, video communication has general functionality as illustrated in Figure 4a, and increasingly includes a link through the air interface as illustrated in Figure 4b. Many digital communication systems and standards, such as MPEG, use coding with variable length codewords for coding efficiency. Variable length decoding (VLD) is needed for decoding bitstreams whenever variable length coding (VLC) is used by the encoder for generating the bitstreams. A VLC table typically has entries with three fields (codeword length: *length*; pattern or information encoded: *pattern*; and variable length codeword: *vlc\_code*). VLD is to determine the value of the fields (*length*, *pattern*) based on the *vlc\_code* value extracted from the bitstream. Figure 2 illustrates the principle of VLD. To find and decode the next codeword, the decoder looks at the sequence of bits forward from the current bitstream position and finds a match to a possible value of *vlc\_code*. Based on the extracted *vlc\_code* value, the VLD determines (*length*, *pattern*) by look up in the VLC table, outputs the value of *pattern* as the decoded next codeword, and then updates the current decoding position in the bitstream according to the decoded codeword *length*, and starts to decode the next codeword. The look for the next codeword usually reads a fixed